

# Busqueda en strings: Boyer-Moore y Knuth–Morris–Pratt

Eduardo Ceja

Algoritmos Avanzados

September 11, 2021

## 1 Booyer-Moore

- Introducción
- Algoritmo

## 2 Knuth Morris Pratt

- Introducción
- Algoritmo

- Desarrollado por Robert S. Boyer y J Stronther Moore en 1977
- El algoritmo de búsqueda de patrones por excelencia
- Es el que se usa en GREP <sup>1</sup>

---

<sup>1</sup>El de GNU

- Ha tenido varias revisiones
- Este algoritmo tiene dos heurísticas
  - Bad character Rule
  - Good Suffix Rule <sup>23</sup>

---

<sup>2</sup>Esta se la pusieron después

<sup>3</sup>Es una mejora de tiempo a costo de memoria

# Bad Character Rule

- Esta regla es simple. El carácter del texto que no sea igual que el de la palabra es el **Bad Character**
- Cuando esto pasa, solo le hacemos shift al patrón hasta:
  - Sean iguales
  - El patrón pase el carácter

# Bad Character Rule

Para el primer caso, hay que

- buscar la posición de la última ocurrencia
- si eso existe en el patrón, solo lo alineamos hasta que sean iguales

# Bad Character Rule

Para el primer caso, hay que

- buscar la posición de la última ocurrencia
- si eso existe en el patrón, solo lo alineamos hasta que sean iguales

# Bad Character Rule

Para el segundo caso:

- Si el carácter no está en el patrón, mueve todo el patrón pasando ese carácter



# Bad Character Rule

Para demostrar el algoritmo, consideremos el siguiente string:  
ANPANMAN y el siguiente patrón: PAN

- ANPANMAN
- PAN

Para demostrar el algoritmo, consideremos el siguiente string:  
ANPANMAN y el siguiente patrón: PAN Ahora empezamos la búsqueda  
de atrás para adelante

- ANPANMAN

- PAN

Como no son iguales, tenemos que seguir la regla, como P si aparece en el patrón, movemos todo a esa posición

Para demostrar el algoritmo, consideremos el siguiente string:  
ANPANMAN y el siguiente patrón: PAN

- ANPANMAN

- PAN

Ya que hicimos el shift, podemos volver a comparar. Son iguales, podemos ver el siguiente carácter

Para demostrar el algoritmo, consideremos el siguiente string:  
ANPANMAN y el siguiente patrón: PAN

- ANPANMAN

- PAN

Ya que hicimos el shift, podemos volver a comparar. Son iguales, podemos ver el siguiente carácter

Para demostrar el algoritmo, consideremos el siguiente string:  
ANPANMAN y el siguiente patrón: PAN

- ANPANMAN

- PAN

Ya que hicimos el shift, podemos volver a comparar. Son iguales, podemos ver el siguiente carácter

Para demostrar el algoritmo, consideremos el siguiente string:  
ANPANMAN y el siguiente patrón: PAN

- ANPANMAN
- PAN

Con esto ya completamos el patrón, por lo que tenemos un match

- Fue creado James H. Morris e independientemente por Donald Knuth
- Vaughan Pratt publico un reporte junto con Morris en 1970
- Lo publicaron el algoritmo en conjunto en 1977

- Este algoritmo, utiliza un preprocesamiento el cual busca los prefijos propios mas largos que también sean sufijos (lps).
- Si tenemos la cadena "AAAB", los prefijos propios serian: "", "A", "AA" y "AAA"
- Lo que se guarda solo es la longitud de estos prefijos,  $\{0,1,2,3\}$
- Esto es para que sepamos a a donde podemos saltar y seguir teniendo matches



Supongamos que tenemos el patron: "AAAA" y el texto: "AAAAABAAA". Primero hay que sacar los lps, los cuales son = {"", "A", "AA", "AAA"}. Ya que se tienen el lps, ya se puede empezar con el algoritmo

Al contrario del algoritmo anterior, este empieza de adelante hacia atrás

- AAAAAABAAA
- AAAA

Como son iguales, lo marcamos como bueno

- A A A A B A A A

- A A A A

Como son iguales, lo marcamos como bueno

- AAABABAAA

- AAAA

Son iguales

- AAAABAAA

- AAAA

Son iguales

Son iguales y ya tenemos un match.

- AAAABAAA
- AAAA

Ahora lo que hay que hacer, es mover el patrón a donde sabemos que ya pasamos uno de los prefijos propios. Esto se logra moviendo el patrón en la longitud anterior al índice donde estamos

- A **AAA** **A** BAAA

- **AAA** **A**

hacen match

Como hacen match, volvemos a hacer shift

- AA**AA**BAAA
- **AAA**A

No hacen match, por lo que tenemos que buscar si hace match con el anterior



- AAAAABAAA

- AAAA

No hacen match, por lo que tenemos que buscar si hace match con el anterior

- AAAAABAAA

- AABA

No hacen match, por lo que tenemos que buscar si hace match con el anterior

- AAAAA**B**AAA

- **A**AAA

No hacen match, pero ya no podemos mover el patrón hacia adelante.